

# Tinyfox

## 80m Foxoring Transmitter

Document version: 5.3  
Date: 5.1.2021  
Authors: Matthias Kühlewein, DL3SDO  
Martin Kuhn, DL3SFB  
English translation: M. Kühlewein, DL3SDO, Lou Crocker, N2END

Hardware version: 5.0  
Software version: 5.3  
Homepage: <http://www.tinyfox.de>

# 1 Introduction

## 1.1 Foxoring

In addition to traditional ARDF, Foxoring is becoming more and more popular. This is an ARDF variant where up to 20 small transmitters are hidden, each of which has a transmission range of about 50 - 200 m. Since the transmitters are located so far away from each other their signals do not overlap, and they transmit their identification continuously on the same frequency.

The participants of a Foxoring get a map where the locations of the transmitters are marked by a circle. However, a transmitter does not need to be exactly at the circle's center or even located inside the circle, but one should be able to receive its transmissions everywhere within the circle.

Due to these characteristics the requirements for 80m Foxoring transmitters are different compared to normal 80m ARDF transmitters:

- low output power
- use of rechargeable or non-rechargeable batteries of low capacity and voltage
- low-cost circuitry
- small size and low weight
- antennas as short as possible
- it is possible to omit a quartz stabilization for the keyer

The **Tinyfox 80m Foxoring Transmitter** described in this document fulfills these requirements and is therefore ideally suited for Foxorings.

## 1.2 Characteristics of the Tinyfox 80m Foxoring Transmitter

### Power supply:

- voltage generation from a single AA cell (either rechargeable or non-rechargeable battery) by a step-up DC-DC converter
- constant output power due to stabilized supply for the RF part
- measurement of battery voltage

### RF part:

- continuously running oscillator
- keyed power amplifier
- antenna matching by a tuned output circuit
- sufficient output power with 30 cm antennas

### Keyer:

- flexible selection of the transmitter identification: letters, numbers, words
- selection of keying speed
- selected identification and keying speed are stored in EEPROM
- possibility of delayed start time up to 59.5 hours
- software is "Open Source" under the GNU General Public License (GPL), therefore you can modify it and adapt it to your own requirements

**Operating the transmitter requires an amateur radio license !**

## 2 Circuit description

The 3.3 V operating voltage is generated by a MAX1724EZK33 step-up converter from a battery voltage within the range of 0.9 ... 5.5 V. The efficiency of the voltage conversion is about 60 % at 1.3 V battery voltage. A permanently transmitting TX consumes 36 mA at this battery voltage. The efficiency of the voltage conversion is mainly influenced by the losses inside C7, C8 und L2, hence it is important to use capacitors with low ESR and coils with low ohmic losses.

The continuously running oscillator consists of a NAND gate and C1, C2, Q1, R1 and R2. The oscillator output is input to three NAND gates connected in parallel, thus capable of driving the required current and acting as power amplifier. The second input of the three gates is used to key the transmitter by the microcontroller. In order to test the transmitter independent of the keying it is possible to drive the input of the gates to high (VCC) by connecting the pins of JP3 (e.g., by a jumper). To avoid a short circuit if the microcontroller attempts to simultaneously drive the input of the gates to low (ground), the microcontroller only actively (low impedance) drives the gates to high. The low-level at the gates results from R6, the microcontroller in this case is high impedance.

The input resistance of the short antenna is predominantly capacitive. The very small ohmic part is almost completely caused by losses and not by the radiation resistance of the antenna. This antenna is, preferably, fed with a high voltage in order to achieve a good range. The series resonant circuit consisting of L1, C6 and the antenna's capacity (a few pF) transforms the output voltage of the power amplifier up and must therefore be tuned to the transmitter frequency. In addition the resulting selectivity at the transmitter's output sufficiently suppresses spurious emissions and harmonics. The maximum attainable voltage at the antenna mainly depends on the quality factor of the inductor. The SMCC inductor used has a Q factor of 40 ... 50 at 3,58 MHz. Within this range of values the output currents of the three parallel connected NAND gates stay within the permissible range. The gate outputs could be overloaded by using a significantly better inductor.

## 3 Set-up

### 3.1 PCB

It is recommended to start with the SMD parts first.

**Important: the voltage converter (IC2) needs to be installed before the tantalum SMD capacitors.**

The voltage converter IC has 5 pins: two on side and three on the other side. It is advisable to first melt some solder on one of the pads at the side where there are only two pads. After carefully locating the IC with suitable tweezers solder the pin to the pad where the tin-solder was put before. Afterwards solder the remaining pins.

**Important: Take care when manipulating the voltage converter IC with tweezers. Grasping the part by the edges may cause it to flip away and be lost if excessive pressure is used.**

Installing the tantalum SMD capacitors is done similar to the voltage converter. First melt some solder on one of the pads. Then locate the capacitor and press it to the pad while soldering the capacitor's metal front side to the pad. Afterwards the other side of the capacitor is soldered. Locate the capacitors centred on the pads since these pads are overlapping with pads provided for an alternative installation of ceramic capacitors of type 1206.

**Important: Mind the polarity! The brown dash indicates the positive connector and shall point towards the voltage converter (C7) resp. be right at the side of the converter (C8). See Layout and Pictures.**

Finally the remaining parts are mounted (keep distance between parts and PCB as close as possible), as well as the solder pins GND, VBAT and ANT. The solder pin GND1 is not mounted, neither are C9 and C10.

For measuring the battery voltage both pins of JP5 need to be connected, either by a jumper or by a soldered wire.

**Important: If the ATtiny (IC1) shall be reprogrammed in-circuit this connection needs to be removed.**

Before putting IC1 (ATtiny) and IC3 (74HC00) into the sockets an electrical test should be performed: connect a fresh AA cell to the GND and VBAT pins (use wires as short as possible to avoid oscillations of the voltage converter) and measure the voltage between JP1 (the pin for the long leg of the LED) and GND. The output of the voltage converter should be 3.3 V.

## 3.2 Mechanical set-up

First the holes for antenna, LED and switch are marked. **Before starting to drill please consider that the housing is made of a smooth plastic material and make sure that the housing is properly clamped.** To get a hole which exactly matches the diameter of the antenna, a pilot hole should be bored first, then reamed to size with a conical hand reamer (which is a very useful tool, especially for smooth plastic material).

Mounting the telescopic antenna (version with white plastic parts, length 141 mm):

Unfortunately the hole inside the angle bracket which will be used to screw the bracket to the housing is covered by the antenna. However, inside the housing there is a circle (resulting from manufacturing the housing), also refer to pictures on the Tinyfox homepage. A 3 mm hole is bored in the center of that circle (which is then 39.5 mm away from the housing's outer edge). Countersink the hole from the outside of the housing and remove plastic retaining parts inside to fix the angle bracket to the housing. The M3 eyelet is mounted between the bracket and an M3 nut (see Pictures).

Before mounting the PCB inside the housing, IC1 and IC3 are put in the respective sockets. After mounting the PCB the eyelet at the bottom of the antenna is connected to the ANT solder pin by a wire as short as possible. Connect the switch to the battery positive terminal as well with the VBAT solder pin by wires as short as possible. Also short the wire for the connection of GND to the negative terminal of the battery. The LED's leads are of different length and are soldered to the pins of JP1 as indicated on the PCB.

The battery holder is glued (e.g., with a hot-glue gun) to the upper part of the housing such that exchanging the battery can be done through the opening of the battery compartment cover. The adhesion can be enhanced by roughening the battery holder and the corresponding surfaces in the housing. **IMPORTANT: Before gluing the battery holder one should check the fit of the holder (by testing the transmitter with an inserted battery) since the wires are not accessible afterwards.**

Optionally a rubber spacer can be glued to the battery compartment cover in order to avoid any risk that the battery does not stay inserted properly inside the holder, e.g., while transporting the transmitter.

The lever of the miniature switch can be shortened to prevent the Tinyfox Transmitter from being switched on unintentionally. Since there is a spiral spring inside the lever that must not jump out, the following procedure is recommended (**at your own risk!**):

1. Carefully notch the lever with a heavy pair of wire cutters round about 3 mm above the external thread.
2. Turn the lever by 90° and notch it with the wire cutters on the same level, then cut it through. **Pay attention that you really turn the lever, since the lever is freely rotatable inside the switch!**

3. Seal the open end of the lever with a small portion of solder - don't solder too long to avoid damage inside the switch. Hint: Quickly add solder until a small ball has formed and then remove the soldering iron immediately.

When mounting the switch be careful not to overtighten the nuts, otherwise the thread will be damaged.

### 3.2.1 Mounting an earthing rod

An earthing rod attached to the Tinyfox transmitter serves two purposes: if used as an electrical ground it can extend the range of the transmitter, and it makes it easier to place the transmitter in the ground without toppling. The mounting of the earthing rod is done as follows:

1. There is a screw hole in the Tinyfox housing, just above the battery case. Drill a 4 mm diameter hole 6.5 mm above the screw hole.
2. A small plate should be placed between the housing and the earthing rod, e.g. a small aluminium plate 22x20x2 mm. Drill a 4 mm hole in the plate 11 mm from each side and 4 mm from one end.
3. Slip an eyelet and a washer on an M4x20 screw and insert the screw through the 4 mm hole from the inside of the housing. Mount the plate on the outside of the housing such that the plate will not cover the screw hole and fix it with an M4 self-locking nut (nylon-insert lock nut).
4. A sheet steel tent peg, e.g. 24 cm long, can be used as earthing rod. Drill a 4 mm hole in the tent peg, e.g. 9 cm away from the upper end. When selecting a tent peg make sure it is sufficiently rounded, such that the self-locking nut fits within the curve and the tent peg lies on the small plate instead of on the nut.
5. The tent peg is slipped on the screw and fixed by an M4 wing nut, which allows for easy mounting and dismounting.
6. If you want an electrical earthing the eyelet is connected to the transmitter's ground, e.g. at the solder pin for the battery ground. By electrically earthing the transmitter the matching is changed. Therefore if the earthing rod is installed after the transmitter was already built up and matched, it needs to be rematched.

## 3.3 Electrical set-up and tuning

Turn the trimmer capacitor (C6) to approx. one third, pull out the antenna completely and switch on the transmitter. The transmitter starts its normal operation by keying for 4s, meanwhile the battery voltage is measured. The battery voltage is then put out twice with a resolution of 10mV. Afterwards the Tinyfox is transmitting its default identification ('A'). For tuning connect the pins of JP3 (with a jumper, e.g.). This forces the transmitter to transmit a continuous tone by overriding the keyer. The tuning of C6 for maximum output power should be done with a non-conductive screwdriver (a properly carved toothpick may also be suitable). When tuning, place the transmitter in a realistic location; if the transmitter is fitted with an earthing rod, it must be inserted into the ground. The transmitted signal can be received by an ARDF receiver 1-2 m away with earphones; for larger distances use a small loudspeaker instead of earphones.

Another method to tune the transmitter is to insert an ammeter into the battery power supply circuit to measure the current. At the maximum current, the output power will be at its maximum. Use leads as short as possible to connect the ammeter. This will avoid oscillations of the voltage converter and will minimize the effect of the ammeter leads on the tuning (since the leads may become part of the antenna system).

## 4 Tinyfox Configuration

All configuration settings like the identification, the keying speed etc. are stored inside the microcontroller's EEPROM.

After programming the microcontroller and after first switching on the Tinyfox transmitter default values are written into EEPROM.

These values can only be changed during **Configuration operation**.

### 4.1 Normal operation

The Tinyfox transmitter is placed in a suitable location by means of the earthing rod. The transmission range can be influenced by varying the length of the telescopic antenna.

After being switched on the Tinyfox starts by transmitting a 4s long continuous tone (if the config push-button is not pressed when the device is switched on and if battery measurement was not disabled), meanwhile the battery voltage is measured.

If a battery warning level is activated, four long dashes are put out if the voltage is sufficient, otherwise 8 dots (= error sign) are put out four times if the voltage is below the threshold.

The measured voltage is then put out one or two times with a resolution of 10mV (e.g., 145 for 1.45V).

Afterwards the identification stored in EEPROM will be transmitted continuously.

EEPROM settings like the identification, keying speed, various operation modes etc. can only be changed or activated during Configuration operation.

In normal operation there won't be any reaction on pressing the config push-button.

### 4.2 Configuration operation

The transmitter must first be opened to get access to the config push-button (S1).

The configuration is solely done by this push-button.

The output happens by morse code being transmitted and signaled via the LED (or being signaled via the LED only).

**Changing the settings (identification and the keying speed etc.) is done in a two-stage process:**

- 1. In the first stage, it is chosen which setting shall be changed**
- 2. In the second stage, the chosen setting is then being changed**

To enter Configuration operation proceed as follows:

1. Switch on the device while keeping the config push-button pressed
2. and release the config push-button within 3.5 seconds. If the the config push-button is not released within these 3.5 seconds the transmitter enters **Normal operation**.
3. The Tinyfox transmitter enters **Configuration operation**.
4. First the version number of the software is put out, e.g. 'V53' followed by a long dash and a pause.

#### **Stage 1:**

In stage 1 it is chosen which setting shall be changed. It starts by putting out number 1, which selects changing the identification. The number is put out three times.

To proceed to the next number (i.e., 2), the config push-button needs to be pressed shortly. If a

number was put out three times without the config push-button being pressed, then stage 2 is entered at which the associated setting will be changed.

## Stage 2:

In stage 2 the setting which was chosen in stage 1 is being changed.

It starts by putting out the actual configured setting.

To proceed to the next possible value, the config push-button needs to be pressed shortly.

If a value was put out three times without the config push-button being pressed, then this setting is written into EEPROM.

The writing to EEPROM is followed by a short pause.

Afterwards the Tinyfox enters **Normal operation** with the new settings.

During outputs in stage 1 and stage 2 the Tinyfox can be switched off at any time.

However, it is recommended not to switch off the device while writing to EEPROM (after third repetition in stage 2). In configuration operation the keying speed is the default speed of 44CpM in stage 1 and 50CpM in stage 2.

Stage 1 Function		Stage 2 Values/settings, details see below	
Config.nr.	Configuration type	Default value/setting	Possible values/settings
1	Identification	'A'	A – Z, 0 – 9, MOE – MO5 etc.
2	Identification keying speed	5 (= 44CpM)	0 – 9 (= 27 – 87 CpM)
3	Battery voltage measurement	2 (= Measurement and comparison against battery warning level 1)	0 – 3
4	Callsign configuration	- empty -	See below
5	Callsign keying speed	5 (= 44CpM)	0 – 9 (= 27 – 87 CpM)
6	Callsign output mode	0 (= no output)	0 – 5
7	Delay time	0 (= no delay)	0 – 59.5h
8	Delay time mode	0 (= not applied)	0 – 2
9	Transmission cycle	0 (= continuously)	0 – 4
10	Position within transmission cycle	1 (= first position)	1 – 5
11	LED deactivation after 5 minutes / Keying of TX during configuration	0 (= No LED deactivation / keying on)	0 – 3
12	Pause between chars	000	0 – 199
13	Battery warning level 1	1.30V	0 – 59
14	Battery warning level 2	1.23V	0 – 59
15	Setting of 12s duration	699	600 – 899
16	Variation 12s duration	3	0 – 4
17	Variation 1min duration	0	0 – 4
18	Variation 5min duration	0	0 – 5

#### 4.2.1 Selecting the identification (1)

The various possible identifications (A–Z, 0–9, =, long dash, error sign, MOE–MO5, MO, MOE\_dash–MO5\_dash, MO\_dash) are put out, starting with 'A'. Pressing the config push-button for at least 2.1s will proceed directly to MOE.

#### 4.2.2 Selecting the identification's keying speed (2)

The various possible keying speeds are put out as numbers from 0 to 9. The relation between the numbers and the (rounded) keying speeds is as follows:

0	1	2	3	4	5	6	7	8	9
27CpM	29CpM	32CpM	35CpM	39CpM	<b>44CpM (Def.)</b>	50CpM	58CpM	70CpM	87CpM

#### 4.2.3 Configuration of the battery measurement (3)

It can be configured if the battery voltage shall be measured after switching on the Tinyfox and how the result shall be put out:

0	No measurement and thus no output
1	Measurement and two times output of the voltage as three-digit number (e.g., 145 = 1.45V)
2	<b>Measurement and comparison against battery warning level 1. If the voltage is below that level then the error sign (8 dots) is put out four times, otherwise a long dash is put out four times. Afterwards the voltage is put out one time. (Default)</b>
3	Same procedure as above at 2, but measurement and comparison against battery warning level 2.

#### 4.2.4 Callsign configuration (4)

It is possible to configure a callsign (or any arbitrary text), which will then be put out. The output mode can be configured by '[Callsign output mode \(6\)](#)'.

The callsign is configured by putting out the selectable chars (alphabetic characters, digits, strings and special chars). Each char is put out three times. Pressing the config push-button shortly will proceed to the next selectable char. If the config push-button is not pressed in time, the char is stored in EEPROM and the next char can be configured.

Pressing the config push-button for at least 2.1s will proceed directly to 0.

The first char is a special char as it represents the end of the callsign. It consists of the 'S' char followed by a long dash. This terminating char is of course not put out when transmitting the callsign in Normal operation.

The second char is also a special char as it represents a pause (as long as 4 dots). It consists of the 'P' char followed by a long dash. Later on a pause is inserted instead of putting out this special char of course. It is possible to configure multiple consecutive pauses.

After this second special char the 'A' char ist put out, followed by 'B', etc.

Since after each selection and storage of a char the next char to begin with is the terminating char, it is not necessary to press the config push-button in order to finalize the callsign, just wait for the terminating char to be put out three times.

The length of the callsign (or arbitrary text) is limited to 80 chars.

#### 4.2.5 Selecting the callsign's keying speed (5)

The various possible keying speeds are put out as numbers from 0 to 9. The relation between the numbers and the (rounded) keying speeds is as follows:

0	1	2	3	4	<b>5</b>	6	7	8	9
27CpM	29CpM	32CpM	35CpM	39CpM	<b>44CpM (Def.)</b>	50CpM	58CpM	70CpM	87CpM

#### 4.2.6 Callsign output mode (6)

<b>0</b>	<b>No callsign output (Default)</b>
1	Only callsign output (instead of the configured identification)
2	Callsign output every 5min
3	Callsign output every 10min
4	Callsign output every 15min
5	Callsign output every 20min

#### 4.2.7 Delay time configuration (7)

The delay time defines the period of time after which the transmitter starts Normal operation.

The possible delay times are put out, beginning with 0 hours (which means configuring no delay time). The delay time is increased by pressing the config push-button: from 0 hours to 1 hour, afterwards in steps of 30 minutes. Delay times up to 59.5 hours can be configured. During configuration the 30min parts are put out as a long dash.

**Please consider that the clock of the microcontroller is derived from the internal RC oscillator instead from an external quartz.**

#### 4.2.8 Delay time mode configuration (8)

It is configurable if and how Delay time operation shall be entered in case of a configured delay time.

<b>0</b>	<b>No Delay time operation (Default)</b>
1	Delay time operation only if the config push-button is being pressed longer than 3.5 seconds

	(or pin 4 and pin 6 at JP2 are connected)
2	Delay time operation

Values 1 and 2 have no effect unless a delay time has been configured.

#### 4.2.9 Transmission cycle configuration (9)

It is configurable at which cycle the configured identification (or the callsign in case of <Callsign output mode = 1>) is put out. This allows a time-controlled operation like for classic ARDF- or sprint transmitters.

**Since the clock of the microcontroller is derived from the internal RC oscillator instead from an external quartz, in practice a time-controlled operation is not possible over a longer period of time.**

<b>0</b>	<b>Continuous transmission (Default)</b>
1	12s transmission duration, 1min cycle (sprint)
2	1min transmission duration, 5min cycle (classic)
3	12s transmission duration, 1min cycle (sprint), started by pressing the config push-button
4	1min transmission duration, 5min cycle (classic), started by pressing the config push-button

#### 4.2.10 Selecting the position within the transmission cycle (10)

It is configurable in which 12s or 1min slot the transmission shall happen if in '[Transmission cycle configuration \(9\)](#)' a cyclic transmission mode has been selected.

The possible values are between 1 and 5.

The default value is 1.

#### 4.2.11 LED deactivation after 5 minutes / Keying of TX during configuration (11)

It is configurable if

- the LED shall be deactivated after 5 minutes of operation (except Configuration operation)
- the output during Configuration operation is only done via the LED or via the LED and by keying the TX (as during Normal operation).

<b>0</b>	<b>No LED deactivation / Output via LED and by keying the TX (Default)</b>
1	LED deactivation after 5 minutes / Output via LED and by keying the TX
2	No LED deactivation / Output only via LED
3	LED deactivation after 5 minutes / Output only via LED

#### 4.2.12 Setting an additional pause between chars (12)

It is configurable if an additional pause shall be inserted between the chars. This can be used for example to increase the difficulty by having longer time intervals between the chars.

The pause is configured by configuring a three-digit number between 1 and 199. By using the config push-button in the usual way the first digit is being configured (either 0 or 1), followed by configuration of the second digit (between 0 and 9), followed by configuration of the third digit (between 0 and 9).

The duration of the additional pause is equal to the configured value multiplied by 0.017152s, a value of 58 thus corresponds to a pause of 1s.

The maximum configurable pause is therefore round about 3.4s.

The default value is 000 (no additional pause between the chars).

#### 4.2.13 Configuration of battery warning level 1 (13)

The battery warning level 1 can be configured that is used if at '[Configuration of the battery measurement \(3\)](#)' the value 2 is selected.

Voltage levels between 1.00V and 1.59V can be selected by configuring a two-digit number between 0 and 59.

By using the config push-button in the usual way the first digit is being configured (between 0 and 5), followed by configuration of the second digit (between 0 and 9).

The default value is 30 (equal to 1.30V).

#### 4.2.14 Configuration of battery warning level 2 (14)

The battery warning level 2 can be configured that is used if at '[Configuration of the battery measurement \(3\)](#)' the value 3 is selected.

Voltage levels between 1.00V and 1.59V can be selected by configuring a two-digit number between 0 and 59.

By using the config push-button in the usual way the first digit is being configured (between 0 and 5), followed by configuration of the second digit (between 0 and 9).

The default value is 23 (equal to 1.23V).

### 4.3 Delay time operation

**Please consider that the clock of the microcontroller is derived from the internal RC oscillator instead from an external quartz.**

After being switched on, the Tinyfox transmitter enters Delay time operation if a [delay time has been configured](#) and if the [delay time mode has been configured](#) appropriately.

Depending on the configuration the Tinyfox enters delay operation immediately or it is first checked whether the config push-button is being pressed longer than 3.5s.

Since the config push-button is connected parallel zu pin 4 and pin 6 of JP6, it is possible to enter delay operation by installing a jumper. The procedure to use delay operation is then as follows:

1. Open the housing and configure the delay time
2. Configure the delay mode
3. Switch off the device
4. Depending on the delay mode configuration: Install a jumper
5. Close the housing
6. After placing the transmitter at its designated location the Tinyfox is switched on and then enters delay operation

It is of course possible to build an external push-button into the housing which is connected parallel to the config push-button.

The delay operation consists of two phases:

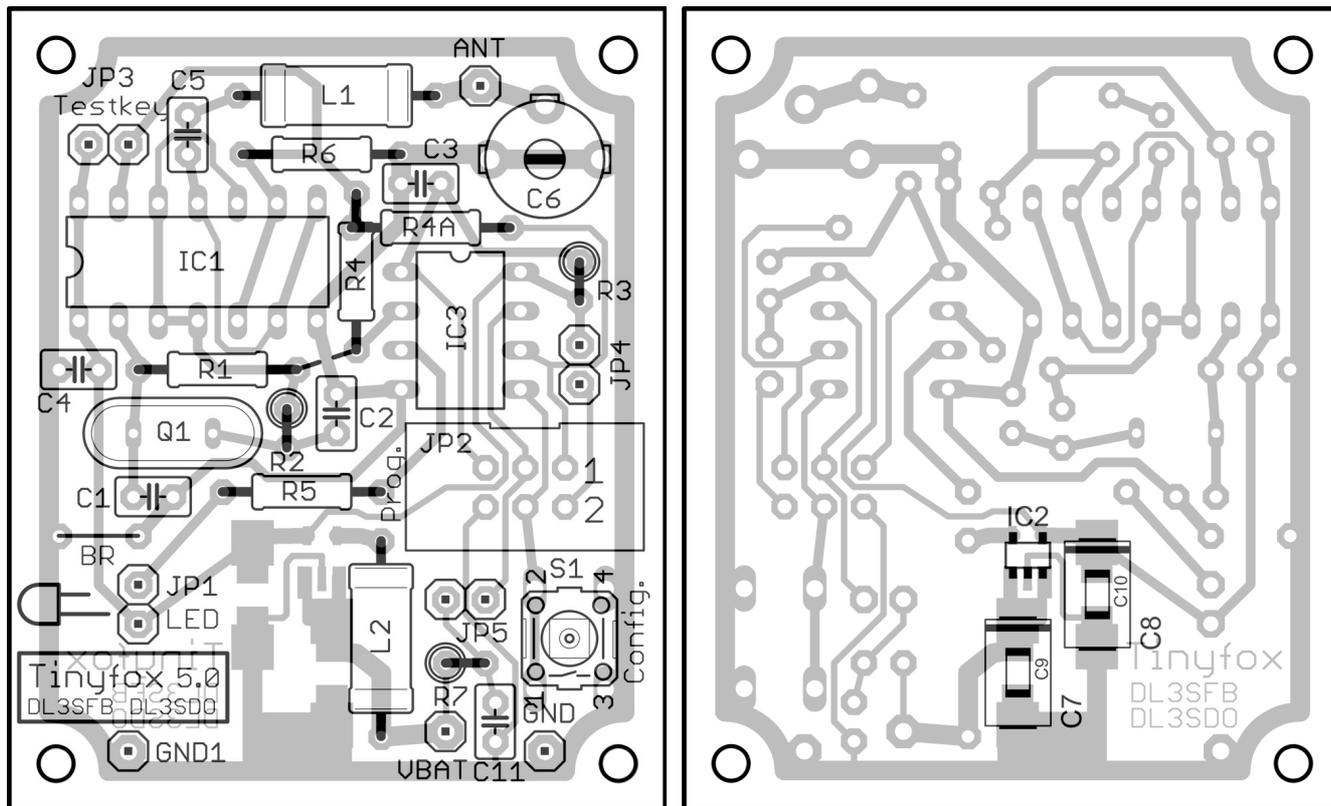
1. Test operation phase: this phase lasts 30 minutes during which the configured identification is being out four times, followed by the error sign and the configured delay time. The test operation phase allows to check the configured identification and to check the transmission range after placement (including overlapping with other transmitters).
2. Waiting phase: this phase follows the test operation phase. Its duration is equal to the configured delay time minus the 30 minutes of the test operation phase. During the waiting phase the transmitter is not being keyed.

After the waiting phase the transmitter enters normal operation.

In delay time operation there won't be any reaction on pressing the config push-button.

# A Layout, part list and schematic

## A.1 Layout



Component side

Soldering side

## A.2 Part list

Parts	Value	Size	Type	Remark
BR1		5	wire jumper	
C1,C2	33p	2.5	ceram.	33
C3,C4,C5, C11	100n	2.5	ceram. X7R	104
C6	2-10p	5 / 7	trimmer cap., 7.5mm Ø	yellow
C7,C8	10µ, 25V	C	low-ESR tantalum SMD	106, 25
IC1	74HC00	14-DIL	CMOS-NAND gate	
IC2	MAX1724EZK33	SOT23-5	step-up converter 3.3V	
IC3	ATtiny25/ATtiny45	8-DIL	microcontroller	ATTINY25 20PU ATTINY45 20PU
JP1,JP3, JP4,JP5		2.5	2-pin header	
JP2		2.5	2x3-pin shrouded box header	
L1	100µ	12.5	SMCC	bn,bk,bn, (gold)
L2	10µ	12.5	SMCC	bn,bk,bk, (gold)
LED	red		low-current, 3mm	assemble in housing and connect to JP1
Q1	3.579MHz	5	quartz HC49-U	3.579545
R1	2.2M	10	¼ watt	rd,rd,gn

R2	1.5K	2.5	¼ watt	bn,gn,rd
R3,R6	10K	2.5,10	¼ watt	bn,bk,or
R4,R5,R7	820	10,2.5	¼ watt	gy,rd,bn
S1		6.5/4.5	miniature push button	

C9, C10 and GND1 are not installed in the current version.

Additional parts:

IC socket 8-pin, IC socket 14-pin, 3 solder pins 1 mm

Mechanical parts:

4 tapping screws 2.2x6.5 mm, 1 countersunk screw M3x6, 2 nuts M3 (low profile, ISO 4035), 1 angle bracket (M3 inside thread in short leg, 3.18 mm hole in long leg, normally used for mounting D-Sub connectors), 1 eyelet M3, 1 battery holder for 1 AA cell, 1 rubber LED-holder for 3 mm LEDs, 1 sub-miniature lever key, 1 telescopic antenna with M3 external thread, 1 plastic housing (SP6060sw) with battery compartment.

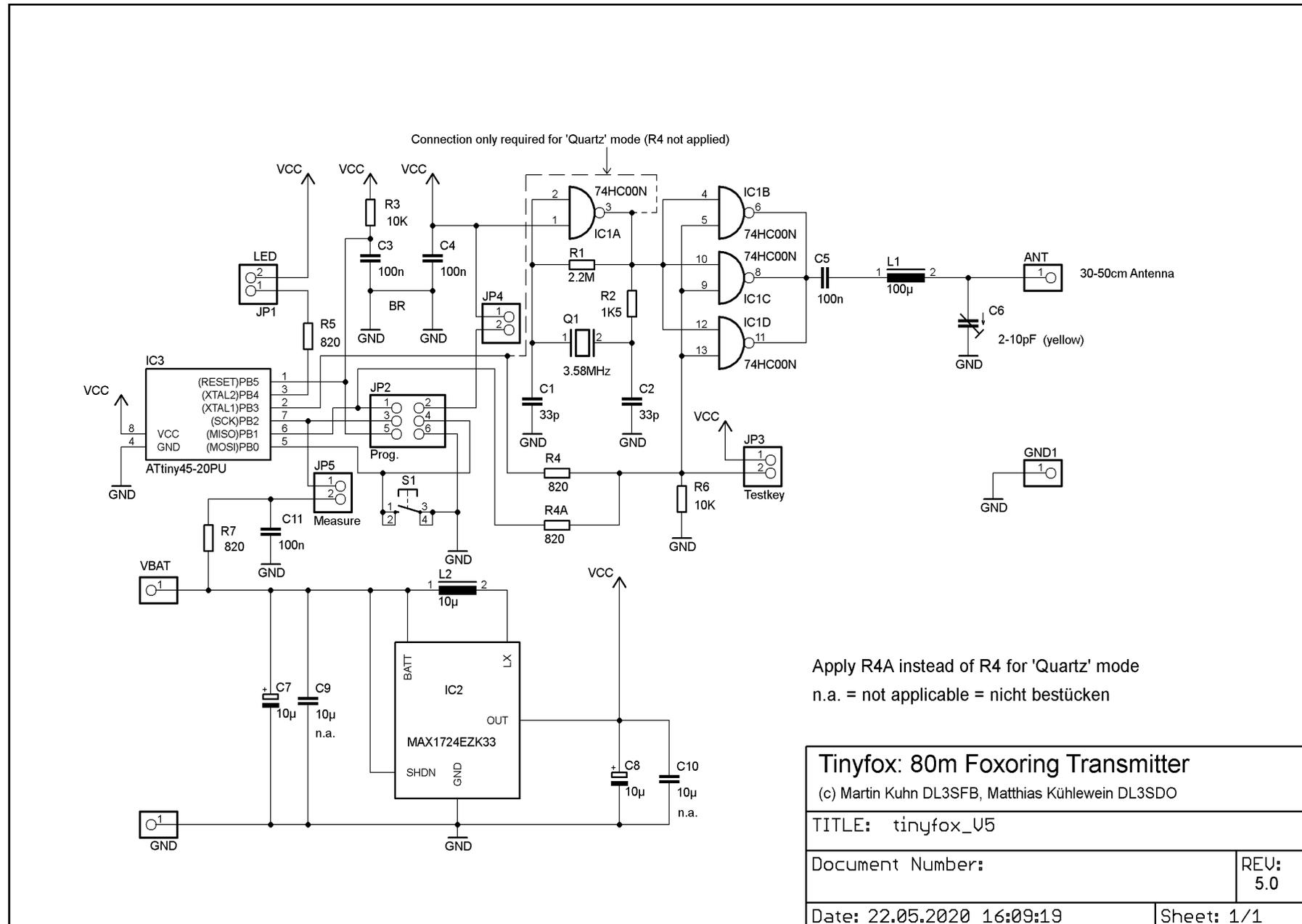
Earthing rod:

1 screw M4x20, 1 eyelet M4, 1 washer M4, 1 small aluminium plate 22x20x2 mm, 1 self locking M4 nut, 1 wing nut M4, 1 sheet steel tent peg 24 cm (e.g., Relags 014800).

Hint:

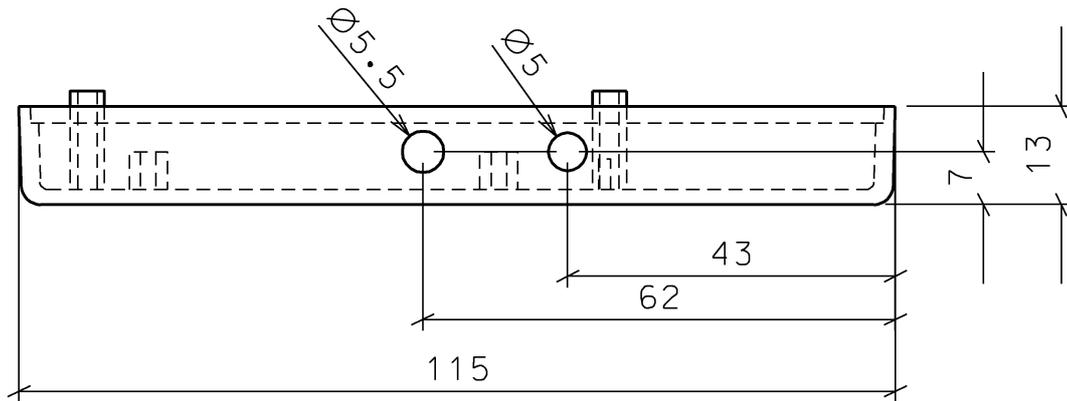
The series resistor R5 can be replaced by a resistor with a larger value which makes the LED light dimmer (and save some energy).

## A.3 Schematic

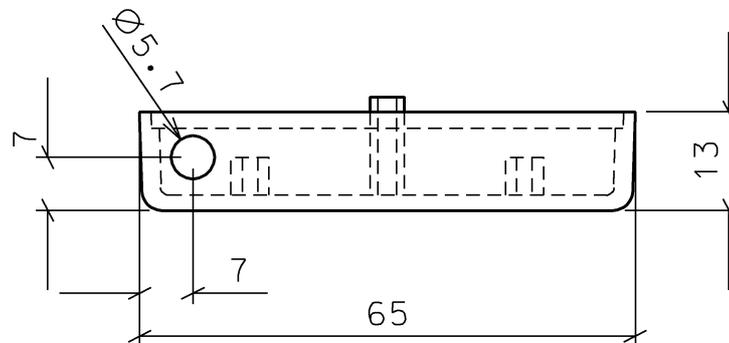


## B Holes

### B.1 LED and switch



### B.2 Antenna



## C Programming the keyer

The microcontroller as part of the kit is already programmed. The following section describes how to modify the software and how to program the device.

### C.1 Programming the $\mu$ C

The software is stored inside the rewriteable flash memory of the microcontroller. Erasing and programming the microcontroller in-circuit is done via its SPI interface which is routed to JP2. There is more than one program for programming microcontrollers. Nowadays, most of them use an USB programmer.

The files required for programming are part of the Tinyfox software package (see Compilation of the software). Compiling the sourcecode is not a prerequisite for programming the microcontroller since the necessary flash files are part of the software package. A description of the various files of the Tinyfox software package can be found in 'readme.txt'.

In most cases it is necessary to connect both pins of JP4 by a jumper to connect pin 2 of JP2 to the 3.3 V supply, otherwise the AVRISP mkII programmer from Atmel (now Microchip) won't work. However, both pins of JP5 must not be connected, which are otherwise connected to measure the battery voltage.

For RC mode, all fuses keep their default state except the Brown-out detection which has to be set to 2.7 V, see <http://www.tinyfox.de/content/programming.htm>

## C.2 Compilation of the software

The zipped archive containing the software can be downloaded from <http://www.tinyfox.de/content/software.htm>. It unzips into the folder 'tinyfox\_software\_Vxx' (xx represents the respective software version).

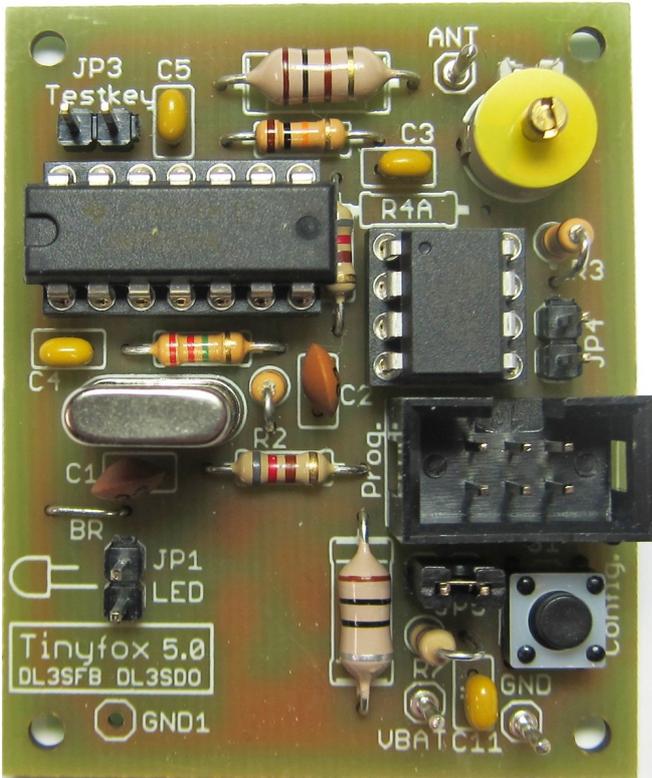
Microchip's AVRASM2 assembler is used to assemble the software. The assembler is not part of the archive. Unfortunately Microchip doesn't offer the assembler as a standalone version. However it is part of the Atmel Studio which can be downloaded from the Microchip homepage (<http://www.microchip.com/avr-support/atmel-studio-7>).

The assembler consists only of the file 'avrasm2.exe' which needs to be copied after installing the Atmel Studio to the 'tinyfox\_software\_Vxx' folder. There are some batchfiles, e.g. 'm\_all.bat', to run the assembly process of the software.

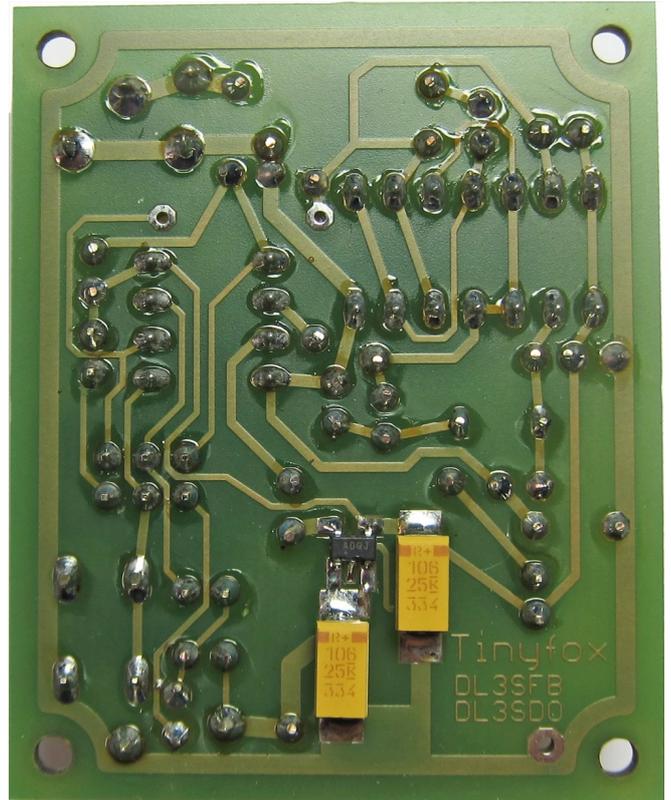
The Tinyfox software (with the exception of the include files 'tn25def.inc' and 'tn45def.inc') is released under the GNU General Public License (GPL), i.e. the software can be modified, copied and distributed under the rules of the GPL. The text of the GPL license is contained within the folder ('COPYING.txt').

The sourcecode of the software only consists of the file 'Tinyfoxxx.asm' (xx represents the respective software version) and the original include files 'tn25def.inc' and 'tn45def.inc' provided by Atmel's former AVR Studio 7.

# D Pictures



Picture 1: Top view (RC mode)



Picture 2: Bottom view (RC mode)



Picture 3: Angle bracket close-up view



Picture 4: Built-up transmitter (PCB V4)